

# A Fine Grain Configurable Logic Block for Self-checking FPGAs

P. K. LALA<sup>a,\*</sup> and A. WALKER<sup>b,†</sup>

<sup>a</sup>Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72701, USA;

<sup>b</sup>Department of Electrical Engineering, North Carolina A&T State University, Greensboro, NC 27411, USA

(Received 15 August 1999; In final form 11 September 2000)

This paper proposes a logic cell that can be used as a building block for Self-checking FPGAs. The proposed logic cell consists of two 2-to-1 multiplexers, three 4-to-1 multiplexers and a D flip-flop. The cell has been designed using Differential Cascode Voltage Switch Logic. It is self-checking for all single transistor stuck-on and stuck-off faults as well as stuck-at faults at the inputs of each multiplexers and the D flip-flop. The multiplexers and the D flip-flop provide either correct (complementary) output in the absence of above-mentioned faults; otherwise the outputs are identical.

**Keywords:** Logic cell; DCVSL; Totally self-checking circuits; FPGAs; On-line testable; Fault-tolerant circuits

## 1. INTRODUCTION

FPGAs are frequently used for rapid prototyping of digital systems [1]. The complexity of FPGAs has increased to an extent that they are used to implement circuits of many thousands of gates. Although FPGAs can be fully tested before a function is programmed into it, faults such as shorts and opens can only be detected after a device has been programmed [2]. Also, the currently available testing techniques can detect only permanent faults, not transient faults that may occur during normal operation [3–8]. The

characteristics of transient faults require a test strategy that is based on continuous monitoring of circuits during normal operation, the presence of a fault being indicated by an invalid output pattern. Such a test strategy is known as concurrent checking or on-line testing. A circuit with concurrent checking capability is known as a self-checking circuit [9]. A typical self-checking circuit shown in Figure 1 consists of a functional part whose outputs are encoded using an error detecting code, and a corresponding checker circuit that monitors the outputs of the functional circuit. A checker must have two outputs and hence, four

---

\*Corresponding author. Tel.: (501) 575-5159, Fax: (501)575-5339, e-mail: lala@engr.uark.edu

†Tel.: (336)344-7761X229, Fax: (336)344-7716, e-mail: alvernon@ncat.edu

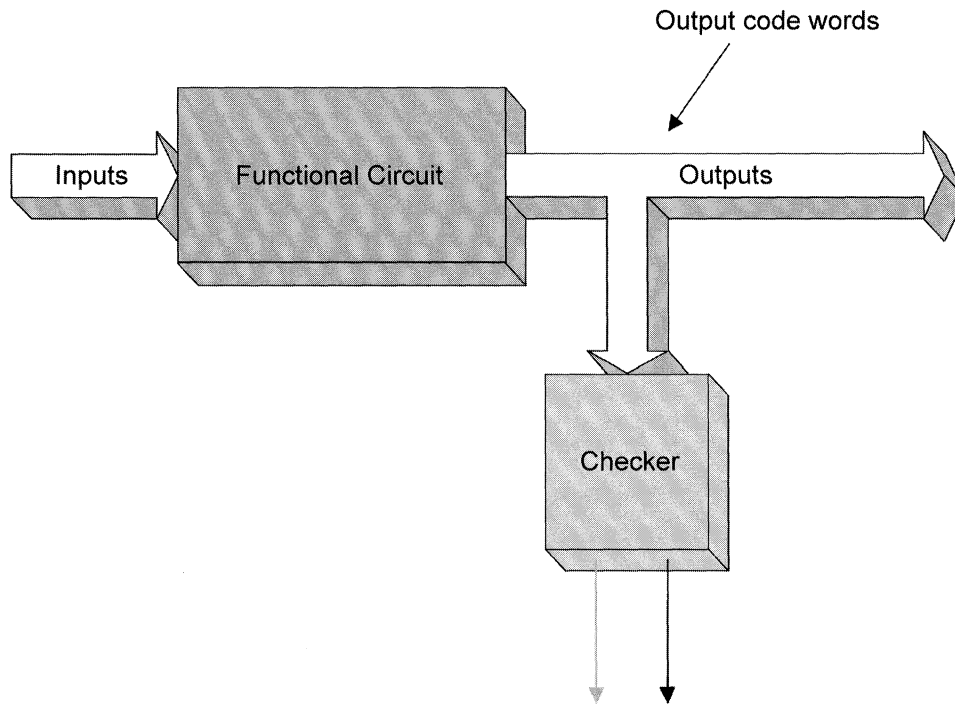


FIGURE 1 Model of a totally self-checking circuit.

output combinations. Two of these combinations are considered to be valid *e.g.* (01 or 10). A non-valid checker output, 00 or 11 indicates either non-code word at the inputs of the checker or a fault in the checker itself [10, 11]. Thus, by observing the output of the checker circuit it is possible to determine whether there is any fault in the function or in the checker circuit.

The following two definitions describe the manner in which self-checking circuits deal with faults [12]:

**DEFINITION 1** A circuit is fault-secure for a given set of faults, if for any fault in the set the circuit either produces the correct code word or a non-code word but never an incorrect code word at the output for the input code space.

**DEFINITION 2** A circuit is self-testing if for every fault from a given set of faults the circuit produces a non-code word at the output for at least one input code word.

A circuit is totally self-checking if it is both fault secure and self-testing. Totally self-checking circuits are very desirable for highly reliable system design since during normal operation all faults from a set would cause a detectable erroneous output. Such circuits have significant advantages, such as:

- (1) Transient faults as well as permanent faults are detected.
- (2) Faults are immediately detected upon occurrence; this prevents propagation of corrupt data within the system.

## 2. SELF-CHECKING CELL

Figure 2 shows the proposed self-checking cell that can be used as a building block for FPGAs. It consists of three 4-to-1 multiplexers, two 2-to-1 multiplexer, and a D flip-flop. The multiplexers

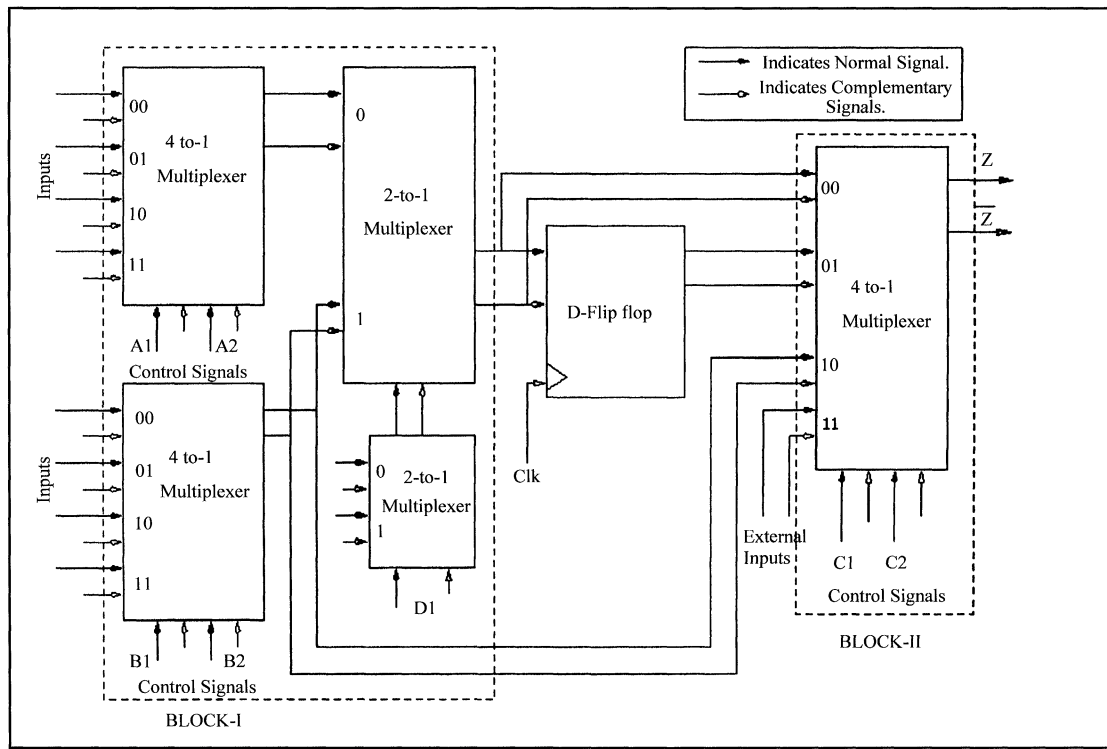


FIGURE 2 Proposed self-checking cell.

and the D flip-flop have been implemented using differential cascode voltage switch logic (DCVSL) [13, 14]. Each cell can implement a given function of upto five variables. Functionally this cell is divided into two main blocks: Block I and Block II. Block I is used to implement the logic function. Block II acts as a control module for the cell.

This module works in four modes:

Mode	Control signal (C <sub>1</sub> , C <sub>2</sub> )	Function
1	00	Five variable combinational function
2	01	Five variable registered function
3	10	Four variable combinational function
4	11	External input

As indicated earlier the proposed cell is implemented using DCVSL. A logic function and its inverse are automatically implemented in this logic

style. The pull-down networks implemented by the nMOS logic tree generated complementary outputs (Fig. 3). The advantage of DCVSL is in its logic density that is achieved by elimination of

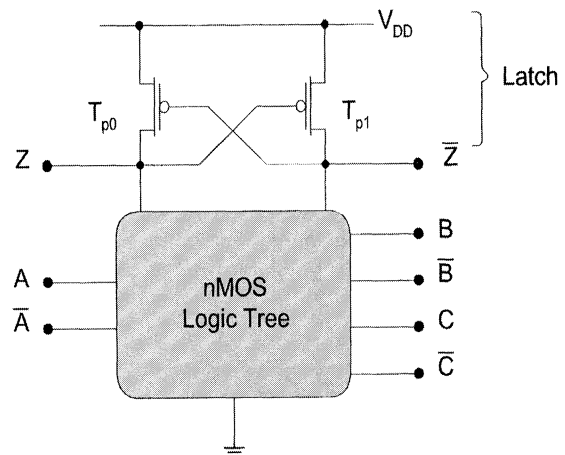


FIGURE 3 General structure of a DCVSL gate.

large PFETS from each logic function. All functions are implemented using NFETS only, and PFETS serve only as the pull-up devices. A DCVSL circuit can be divided into two basic parts: a differential latching circuit and a cascoded complementary logic array. The latch in these logic circuits is realized with two cross-coupled pMOS transistors as shown in Figure 3, *i.e.*, transistors Tp0 and Tp1. The cascode complementary logic array is realized with a nMOS logic tree. The function is realized by the part of the nMOS logic tree connected to node Z. The complement of that function is realized by the section of the logic tree connected to node  $\bar{Z}$  in Figure 3. Due to the two outputs (or lines) associated with dual-rail logic, four distinct signal states are possible. To illustrate the behavior of DCVSL circuits in the normal and single stuck-at faults case we consider the 2-to-1 multiplexer. Figure 4 shows a 2-to-1 multiplexer with inputs A and B and their complements  $\bar{A}$  and  $\bar{B}$  respectively; S and  $\bar{S}$  are the control signals. Table I shows the normal operation of the multiplexer. Control signal S is high ( $S=1$ ) and  $\bar{S}$  is low ( $\bar{S}=0$ ) for vectors 1 to 4. For vector 1, input  $A=1$ ,  $\bar{A}=0$ ,  $B=1$  and  $\bar{B}=0$ , the output response are  $Z=1$  and  $\bar{Z}=0$ .

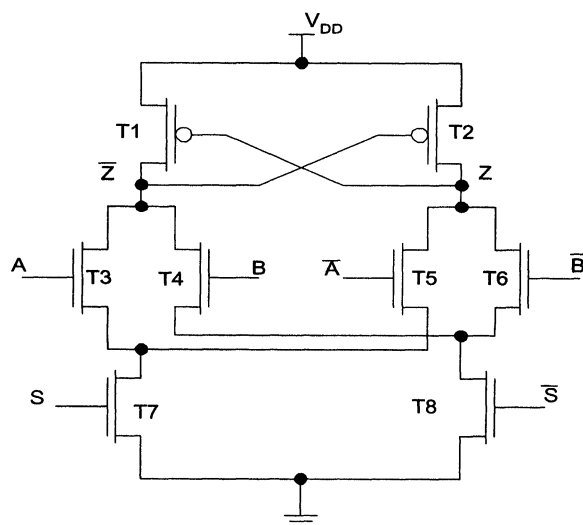


FIGURE 4 DCVSL 2-to-1 multiplexer.

TABLE I Fault free operation of 2-to-1 multiplexer in Figure 4 for a stuck-at 0

Vectors	S	$\bar{S}$	A	$\bar{A}$	B	$\bar{B}$	Z	$\bar{Z}$
1	1	0	1	0	1	0	1	0
2	1	0	1	0	0	1	1	0
3	1	0	0	1	1	0	0	1
4	1	0	0	1	0	1	0	1
5	0	1	1	0	1	0	1	0
6	0	1	0	1	1	0	1	0
7	0	1	1	0	0	1	0	1
8	0	1	0	1	0	1	0	1

Similarly, vector 2 to 4 during normal operation produce either 01 or 10 outputs. Vectors 5 to 8 are similar to vectors 1 to 4 except for control signal  $S=0$  and  $\bar{S}=1$ . Now let us examine the behavior of the circuit shown in Figure 4 for a stuck-at 0 fault on input A with  $A=0$ ,  $\bar{A}=1$ ,  $B=0$ ,  $\bar{B}=1$ ,  $S=1$  and  $\bar{S}=0$ . Both T7 and T8 are off for this input. Therefore both pulldown networks are off in this case. In this state the output of the DCVSL gate is only determined by the differential latching circuit, *i.e.*, transistors T1 and T2 in Figure 4. The output of the circuit corresponds to the two states of the latch, *i.e.* ( $Z\bar{Z}$ ) = 01 and 10, the output state of the circuit, *i.e.*, the state of the latch, is set during the last fault-free event in the circuit. The output of the DCVSL gate should be 11 or 00 in this case for online fault detection. Table II shows the behavior of the 2-to-1 multiplexer in the presence of a s-a-0 fault at its inputs for a previous circuit output of 01, *i.e.*,  $Z=0$  and  $\bar{Z}=1$ . Vectors 1 and 2 in Table II indicates the behavior of this multiplexer when input A is s-a-0. The desired response of the DCVSL circuit is also shown in Table II. Both pulldown networks are not conducting for all the vectors and s-a-0 faults in Table II. In this mode the output of the gate is determined by the differential latch. The output of this circuit doesn't indicate the presence of a fault because the two states of the differential latch 01 or 10 correspond to valid output codes, *i.e.*, 01 or 10. To indicate the presence of a fault it should be either 11 or 00. If we modify the 2-to-1 Multiplexer by adding two weak p channel devices, *i.e.*,

TABLE II Behavior of 2-to-1 multiplexer in the presence of single stuck-at-0 faults

Vectors	S	$\bar{S}$	A	$\bar{A}$	B	$\bar{B}$	S-a-0 fault	Circuit response		Modified circuit response	
								Z	$\bar{Z}$	Z	$\bar{Z}$
1	1	0	1	0	1	0	A	0	1	1	1
2	1	0	1	0	0	1	A	0	1	1	1
3	1	0	0	1	1	0	$\bar{A}$	0	1	1	1
4	1	0	0	1	0	1	$\bar{A}$	0	1	1	1
5	0	1	1	0	1	0	B	0	1	1	1
6	0	1	0	1	1	0	B	0	1	1	1
7	0	1	1	0	0	1	$\bar{B}$	0	1	1	1
8	0	1	0	1	0	1	$\bar{B}$	0	1	1	1
9	1	0	1	0	1	0	S	0	1	1	1
10	0	1	0	1	0	1	$\bar{S}$	0	1	1	1

transistors T9 and T10, as shown in Figure 5. The output of the modified 2-to-1 Multiplexer will equal 11 for all the vectors and s-a-0 faults in Table II. This will also be its Figure 5 here output for any single fault that causes both pulldown networks to be nonconductive. The DCVSL rise time characteristics is determined by the geometry of the weak p-channel pullup transistors, *i.e.*, T9 and T10 in Figure 5. The propagation and power

dissipation of the Multiplexer is also affected by this configuration. Stuck-at faults can also cause both sides of the logic tree to conduct simultaneously.

A list of vectors and stuck-at faults that causes the circuit to operate in this mode is given in Table III. Table III shows the expected outputs for all single stuck-at-1 faults at the inputs of 2-to-1 multiplexer. The input vectors are the same as vectors 1 and 2 of Table I and Table II but both Z and  $\bar{Z}$  should become 0 in the presence of an input s-a-1 fault. To determine the output of this dual-rail circuit when both pulldown networks conduct consider the DCVSL inverter shown in Figure 6 for an input of  $V_{DD}$  on both gate inputs. If we assume that the threshold voltage of the transistors in this circuit are equal and that  $\beta_1/\beta_4 = \beta_2/\beta_3$  then

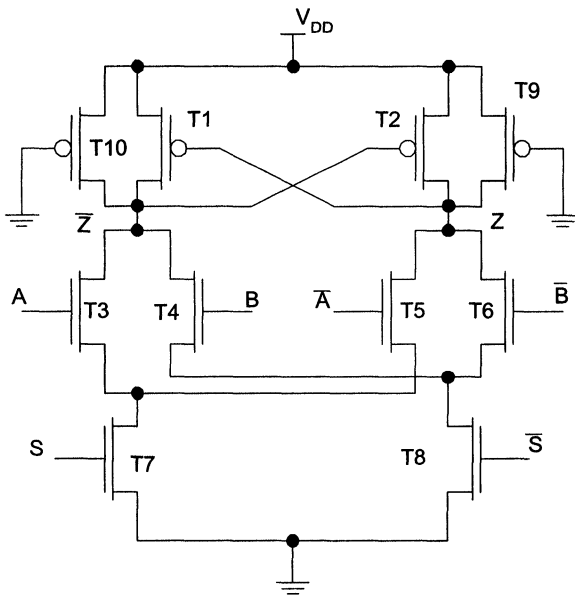


FIGURE 5 Modified DCVSL 2-to-1 multiplexer.

TABLE III Output response for Stuck-at-1 conditions

Vectors	S	$\bar{S}$	A	$\bar{A}$	B	$\bar{B}$	S-a-1 fault	Z	$\bar{Z}$
1	1	0	1	0	1	0	$\bar{A}$	0	0
2	1	0	1	0	0	1	$\bar{A}$	0	0
3	1	0	0	1	1	0	A	0	0
4	1	0	0	1	0	1	A	0	0
5	0	1	1	0	1	0	$\bar{B}$	0	0
6	0	1	0	1	1	0	$\bar{B}$	0	0
7	0	1	1	0	0	1	B	0	0
8	0	1	0	1	0	1	B	0	0
9	0	1	1	0	0	1	S	0	0
10	1	0	0	1	1	0	$\bar{S}$	0	0

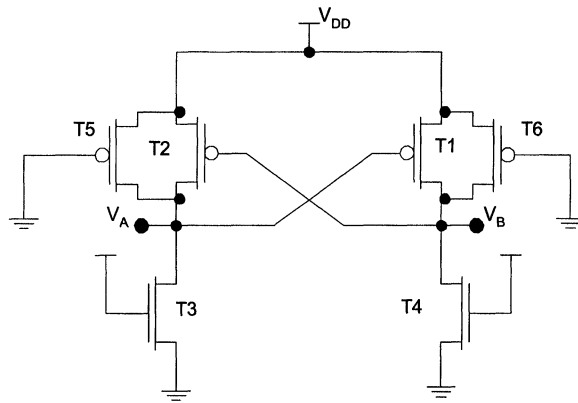


FIGURE 6 DCVSL inverter.

the voltage on nodes  $V_A$  and  $V_B$  is,

$$V_B, V_A, (V_{DD} \times V_{TN}) \left\{ 1 \times \sqrt{\frac{\beta_3 \times \beta_5}{\beta_3 \% \beta_2}} \right\}$$

Therefore the ratio of the MOS transistor gain factor  $\beta$ , for transistors  $T_1$ ,  $T_2$ ,  $T_3$  and  $T_4$ , that is required to insure that the output of the gate falls within the output low noise margin  $NM_L$  is defined by the following relationship,

$$\frac{\beta_3 \times \beta_5}{\beta_3 \% \beta_2} \beta \left( \frac{V_{DD} \times V_{TN} \times V_{OL_{max}}}{V_{DD} \times V_{TN}} \right)^2$$

where,

$V_{OL_{max}}$  — maximum LOW output voltage.

Table III shows the outputs for all single stuck-at-1 faults at the inputs of the modified 2-to-1 multiplexer. The input vectors are the same as vectors 1 and 2 of Table I and Table II but both  $Z$  and  $\bar{Z}$  become 0 in the presence of an input s-a-1 fault. Transistor stuck-on or stuck-off faults have similar effect on the outputs as s-a-1 and s-a-0 faults respectively. Some of which are shown in

TABLE IV Output response for stuck-at-1 conditions

S	$\bar{S}$	A	$\bar{A}$	B	$\bar{B}$	Fault	Z	$\bar{Z}$
1	0	1	0	1	0	T1 stuck-on	1	1
1	0	1	0	1	0	T2 stuck-off	0	0
1	0	0	1	1	0	T3 stuck-on	0	0
0	1	1	0	0	1	T4 stuck-on	0	0
1	0	0	1	1	0	T5 stuck-off	1	1
0	1	1	0	0	1	T6 stuck-off	1	1
1	0	1	0	1	0	T7 stuck-off	1	1
1	0	1	0	0	1	T8 stuck-on	0	0

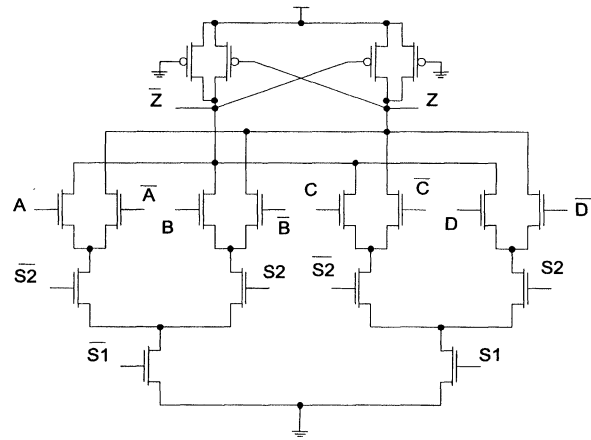


FIGURE 7 A Self-checking 4-to-1 multiplexer.

Table IV. The 4-to-1 multiplexer behaves in the same manner as the proposed 2-to-1 multiplexer in the presence of all-single stuck-at faults and transistor stuck-on/off fault *i.e.*, the transient response for normal and s-a-1/0 will show the similar pattern for all defined fault-free and faulty conditions. Its transistor level diagram is shown in Figure 7. Figure 8 shows the Self-checking master-slave D-flip-flop that has inputs Clk (clock), D (data-in) and their complements  $\bar{C}lk$  and  $\bar{D}$  respectively; the outputs are  $Q$  and  $\bar{Q}$ . The fault-free operation of this D-flip-flop is shown as vector 1 in Table V. Vectors 2 to 6 show the behavior of D-flip-flop in the presence of assumed stuck-at-faults. For any transistor-fault or faulty input conditions *i.e.*, 00 or 11,  $Q$  and  $\bar{Q}$  either provides correct output or non-code word.

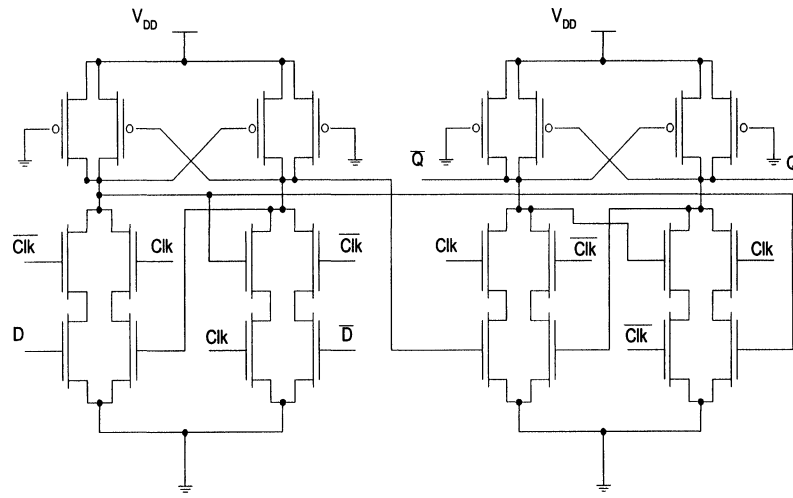


FIGURE 8 Self-checking D flip-flops.

TABLE V Operation of D-flip-flop

Vectors	D	$\bar{D}$	Clk	Clk	Fault states	Q	$\bar{Q}$
1	0	1	1	0	No Fault	1	0
2	0	1	1	0	D s-a-1	0	0
3	0	1	0	1	$\bar{D}$ s-a-1	0	0
4	1	0	1	0	D s-a-0	1	1
5	1	1	1	D and $\bar{D}$ s-a-1	$\bar{D}$ s-a-1	0	0
6	0	0	1	D and $\bar{D}$ s-a-0	$\bar{D}$ s-a-0	1	1

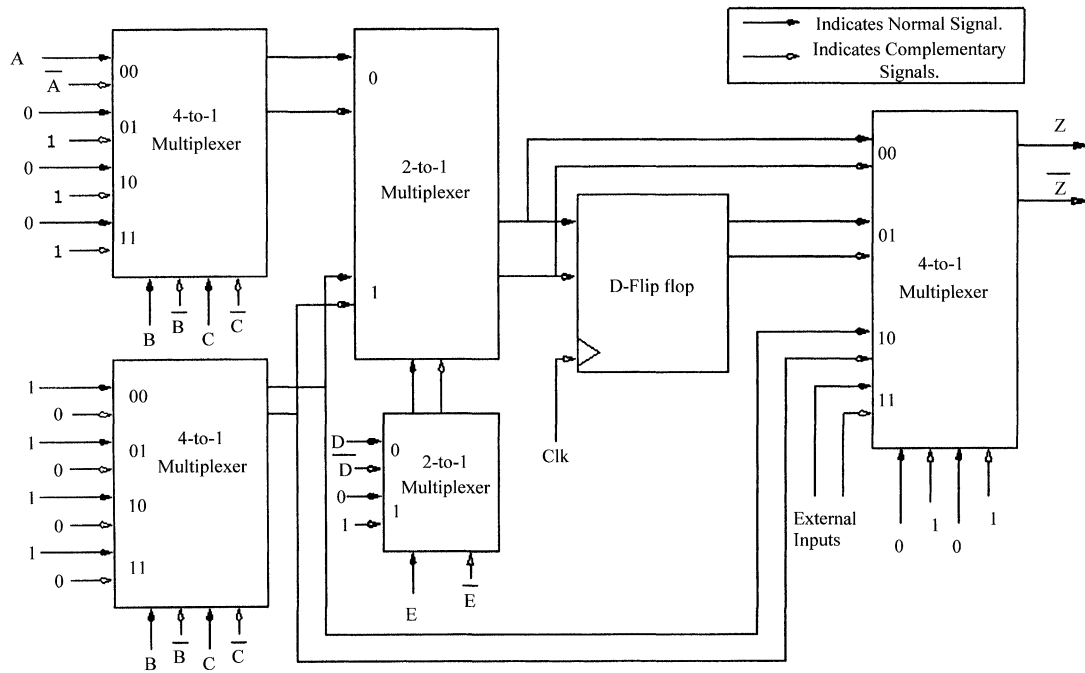


FIGURE 9 Implementation of example on proposed cell.

### 3. APPLICATION

To illustrate the application of the proposed cell in self-checking logic design let us consider an

expression of five variables:

$$Z = A\bar{B}\bar{C} + D\bar{E}$$

This can be illustrated by using proposed cell as shown in Figure 9. Let us assume A is stuck-at-1 then for input pattern  $A=0, B=0, C=0$  and  $E=1$ , outputs are  $Z=0$  and  $\bar{Z}=0$ . If A is stuck-at-0 then for input pattern  $A=1, B=0, C=0$  and  $E=1$  outputs are  $Z=1$  and  $\bar{Z}=1$ . Similarly, for other single stuck-at fault in this cell Z and  $\bar{Z}$  gives either the expected output or the wrong output. Output waveforms (generated by Berkeley SPICE-3.5) for the fault-free cell, A stuck-at-0, and A stuck-at-1 condition are shown in Figures 10, 11 and 12 respectively. In Figures 10, 11 and 12, nodes  $\nu(9)$  and  $\nu(8)$  represent output Z and  $\bar{Z}$  of the circuit shown in Figure 9. Figure 13 shows the layout of proposed cell that has been implemented using Magic layout in  $2\text{-}\mu$  CMOS technology. We illustrate the implementation of the seven MCNC benchmark circuits using the proposed cell. They are tested in the presence of faults and have produced the similar results as derived in fault tables. These are listed in Table VI. It is also found

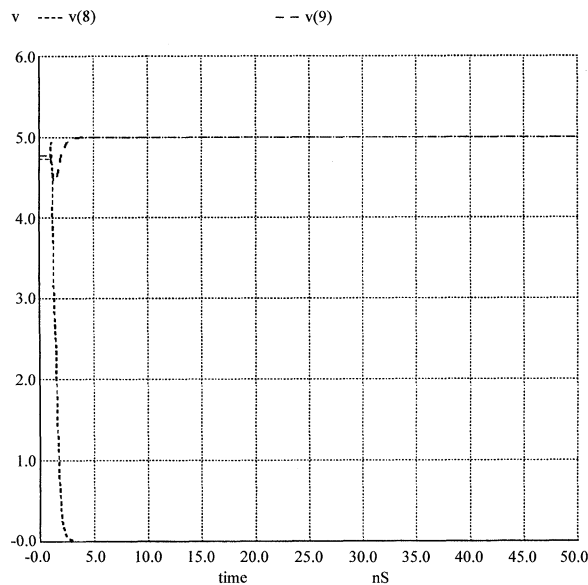


FIGURE 10 Fault-free condition.

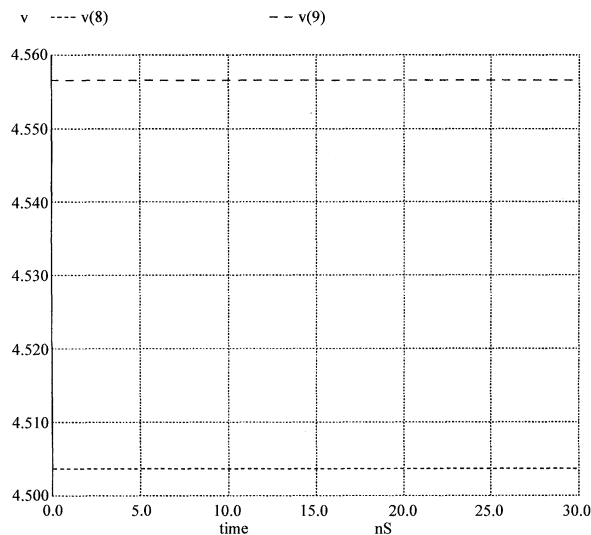


FIGURE 11 Stuck-at-0 condition.

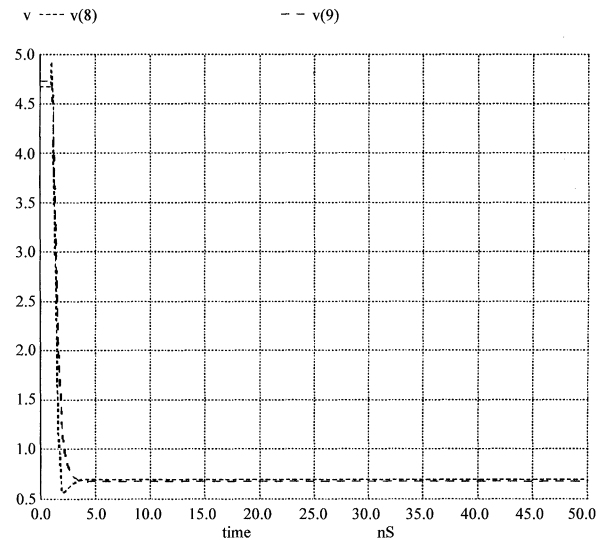


FIGURE 12 Stuck-at-1 condition.



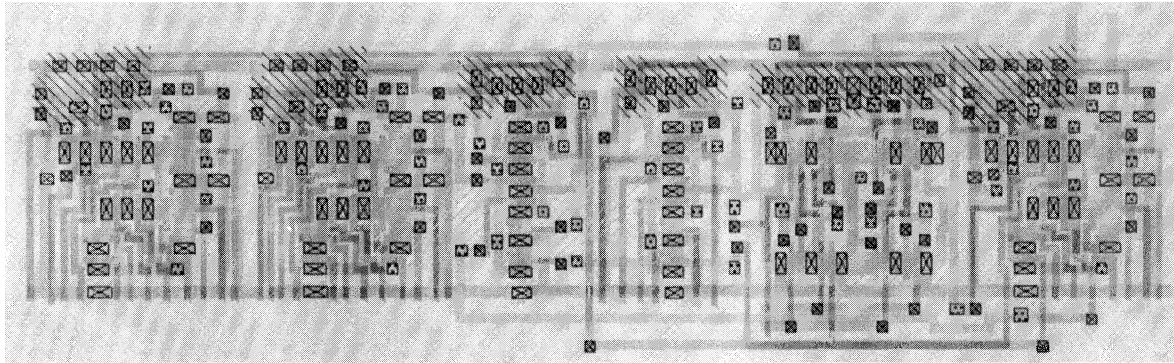


FIGURE 13 Layout of self-checking cell (Size:  $4358 \times 908$  in  $2\text{-}\mu$  technology).

TABLE VI Implemented benchmarks using proposed cell

	MCNC benchmark	# of cells
1	apex2	1
2	Cordic	1
3	rd84	1
4	sao2	1
5	vg2	2
6	5expl	3

that most of the complex functions required only one cell except for vg2 and 5expl, which are eight and ten variable functions respectively.

#### 4. CONCLUSION

We have developed a programmable cell with built-in self-checking feature. This cell can be used as a configurable logic block in an FPGA. The major features of such an FPGA are:

- (1) Each configurable cell uses fewer transistors than that required in FCMOS implementation.
- (2) A single transistor stuck-on or stuck-off fault in a multiplexer or in the D flip-flop will result in an output of 00 or 11 from the cell. Also, any single stuck-at fault at the inputs of a multiplexer or the D flip-flop can be detected on-line.
- (3) A number of such cells can be interconnected to implement any complex logic function. If

such cells are interconnected in  $n$ -stages, the presence of a single fault in one of the intermediate cells will propagate to the output of the final stage. The final outputs of such a function can be verified internally by incorporating two-rail checkers.

#### References

- [1] Salcic, Z. and Smailagic, A., *Digital Systems Design and Prototyping using Field programmable Logic*, Boston: Kluwer Academic Publishers, 1997.
- [2] Burruss, A. L. and Lala, P. K., "On-line testable logic design for FPGA implementation", *Proc. 1997 International Test Conference*, pp. 471–478.
- [3] Lala, P. K. and Missen, J. I., "Method for the diagnosis of a single intermittent fault in combinational logic circuits", *Proc. IEEE*, 1979, pp. 187–190.
- [4] Koren, I. and Kohavi, Z., "Diagnosis of intermittent faults in combinational networks", *IEEE Trans. Comput.*, Nov. (1977), pp. 1154–1157.
- [5] Savir, J. (1980). "Detection of single intermittent faults in sequential circuits", *IEEE trans.*, pp. 673–678.
- [6] McCluskey, E. J. and Wakerly, J. F. (1981). "A circuit for detecting and analyzing temporary failures", *Proceedings of IEEE COMCON*, pp. 317–321.
- [7] Stifler, J. I. (1980). "Robust detection of intermittent faults", *Proceedings of IEEE international symposium on Fault tolerance computing*, pp. 216–218.
- [8] Hurst, Stanley L., *VLSI Testing: digital and mixed analogue/digital techniques*, IEE 1998.
- [9] Lala, P. K., *Fault Tolerant and Fault Testable Hardware Design*, Englewood Cliffs: Prentice-Hall, 1985.
- [10] Anderson, D. A. and Metze, G., "Design of totally self-checking check circuits for  $m$ -out-of- $n$  codes", *IEEE Trans. Comput.*, March, 1973, pp. 263–269.
- [11] Gastanis, N. and Halatsis, C., "A new design method for  $m$ -out-of- $n$  TSC checkers", *IEEE Trans. on Comput.*, March 1983, C-32(3), 273–283.

- [12] Nicolaidis, M., "Fail-Safe Interfaces for VLSI: Theoretical Foundations and Implementation", *IEEE Trans. Comput.*, Jan., 1998, pp. 62–77.
- [13] Jan M. Rabaey, *Digital Integrated Circuits; a design prospective*, Upper Saddle River: Prentice-Hall, 1996.
- [14] Heller, L. G. *et al.*, "Cascode Voltage Switch Logic: A differential CMOS Logic Family", *Proceedings of 1984 IEEE International Solid-state Circuits Conference*, pp. 16–17.

### Authors' Biographies

**Parag K. Lala** is the Mullins Chair Professor in the Department of Computer Science and Computer Engineering, University of Arkansas. He received an M.Sc. (Eng.) degree from King's College, London, a Ph. D. degree from the City University of London, and a D.Sc. (Eng.) degree from the

University of London. He is the author/co-author of more than 100 publications. He is also the author of five books. His new book *Self-checking and Fault Tolerant Digital Design* has been recently published by Morgan-Kaufmann. He is an Associate Editor of IEEE Trans. on VLSI Systems.

**Alvernon Walker** is an Associate Professor in the Department of Electrical Engineering at North Carolina A&T State University. His research interests include mixed-signal system test generation/testability, Mixed-signal BIST, VLSI Mixed-signal system design, and self-timed system design. He holds a BS and an MS in electrical engineering from North Carolina Agricultural and Technical State University and a Ph. D. in Electrical Engineering from North Carolina State University.

